Nikolaos Kranidiotis – OSEC.GR

Vehicle & IoT Forensics Cheat Sheet

1. Quick Triage (live handling)

- Document everything upon arrival: date/time, vehicle status (ignition on/off), screens (navigation display, dash info) – photograph or note volatile info before it disappears. Ensure any connected mobile devices or key fobs are identified and secured.
- Minimize changes to the system: avoid unnecessary power cycles (each wake/sleep can purge volatile memory). If the vehicle is on, do not turn it off until volatile data (e.g. current route, live telematics) is captured or documented.
- If vehicle is running, consider a controlled shutdown: turn off ignition and remove keys, close all doors, wait a few minutes for ECUs to enter low-power mode, then disconnect battery (or use OEM transport mode) to preserve state. Verify the vehicle fully powered down (screens & lights off) before moving it.
- Isolate from networks immediately: remove SIM cards from telematics modules, unplug or shield antennas (cellular, Wi-Fi, Bluetooth). Use Faraday bags for key fobs or IoT devices to prevent remote wipe signals. (If a device must remain on for memory preservation, isolate radio communications instead of powering off.)
- Secure external storage and peripherals: check for USB drives, SD cards, or paired smartphones in the vehicle – these can contain additional evidence and should be collected promptly. For IoT devices, also collect any hub or base station.
- For IoT gadgets at a scene, cut power or network access to prevent data loss: unplug devices or remove batteries (if it won’t erase data) and isolate them. If a device can’t be powered down safely, place it in a Faraday container to block signals (do not enclose loose lithium batteries in Faraday material for long).
- Maintain chain-of-custody from the start: label each component (ECUs, cables, storage media) and log the person, time, and method of collection. Handling should align with agency policy – gloves for physical evidence, anti-static bags for electronics, tamper-evident seals on removed modules, etc.

2. Evidence Sources & Devices

- **Vehicle systems:** Infotainment head units and telematics control units are primary sources (user data, logs). Also consider ECU modules (engine, transmission, body control) for fault logs, and the airbag control module (contains EDR crash data).
- **Aftermarket add-ons:** OBD-II dongles (usage-based insurance or fleet trackers), GPS trackers, dash cameras, and immobilizer/alarm systems can all store trip data or events. These should be located and seized (e.g., look under dashboard for OBD devices or hardwired trackers).
- **Connected mobile devices:** The driver's phone (paired via Bluetooth/USB) holds call logs, app data (e.g. navigation searches) and cloud sync info that might not reside in the car. Likewise, if Android Auto or CarPlay was used, the handset will have relevant data (messages, music history) – preserve it via standard mobile forensics.
- **IoT devices:** Smart home appliances (voice assistants, security cameras, thermostats, smart locks, wearables) can be evidence. Identify all IoT in the environment: they often work with companion apps or cloud services. Collect the devices themselves *and* associated items (smart hubs, memory cards, linked phones, account credentials).
- **External/cloud sources:** Vehicle manufacturer servers (e.g., Tesla, OnStar) and IoT cloud platforms store logs and telemetry. Plan to obtain these via legal process. For example, GM OnStar records remote horn honks, location pings; Tesla logs driving and Autopilot data to its cloud. Don't forget traffic cameras, toll records, and other external data that can complement digital evidence.
- Every module with storage is an evidence source: infotainment flash memory, SD maps, SIM cards, even key fobs (which may log last usage). Create an inventory of all such components in the vehicle and on IoT networks, so none are overlooked during collection.

3. Common File Systems & Data Formats

- **Automotive file systems:** Many infotainment systems run embedded Linux or QNX. Expect formats like ext4 (common in Android-based systems), QNX6 FS (proprietary, found in many OEM head units), FAT32/exFAT for SD cards or USB storage, and sometimes NTFS or proprietary for certain media systems. *Example (model/version-specific):* some FCA/Chrysler Uconnect QNX systems expose internal flash under `/fs/mmc0/`.
- **IoT device storage:** IoT devices often use flash-friendly file systems: YAFFS2, JFFS2, or UBIFS on raw NAND; smaller devices may not have a formal FS (just key-value storage in firmware). If the device runs Linux, ext3/ext4 are possible on internal eMMC. Extract full images when possible and be prepared to handle unusual file systems with custom tools or mount in an emulator.
- **Data formats:** Expect plenty of SQLite databases (contacts, calls, messages, app caches) in infotainment and companion apps. XML/JSON and binary log files are common for telematics and IoT sensor data. Navigation systems may use GPX, KML, or proprietary POI files for saved locations. CAN bus diagnostic data could be in DTC codes (stored in hex) or "freeze frame" reports. Always check for timestamps in Unix epoch, CAN timestamps (relative), or GPS time formats.
- **Examples:** Phonebook contacts might reside in `/fs/mmc0/app/share/` databases on a QNX unit; vehicle event logs could be in CSV-like logs (e.g., Ford SYNC generates .TXT logs of events). Smart home devices might log to `/var/log/` or a SQLite DB (`events.db`) internally. Knowing the typical format (e.g., Nest cameras use .mp4 clips, Alexa stores voice transcripts as text) helps focus your search.
- Use specialized tools or modules for obscure file systems: The Sleuth Kit/Autopsy has plug-ins for many FS; for QNX, consider QNX6 FS drivers or carve with Photorec if direct support is lacking. For raw flash, tools like binwalk (to find embedded filesystems) and bulk extractor (to find remnants of known data structures) are valuable.
- Be mindful of endianness and encoding: certain automotive binary files may store multibyte values in big-endian, or use non-standard character encoding. When parsing proprietary formats, refer to any available documentation or community research to decode them correctly.

4. Core Artifacts & Locations (per device class)

- **Infotainment Head Unit:** Stores user phone data (downloaded contacts, call logs, SMS messages) when phones are paired. Contains navigation data (favorites, recent destinations, last GPS position) often in nav databases. Check for media files or metadata (music playlists, paired Bluetooth device list). Typical storage in internal flash; artifacts often under directories like `/nav/`, `/contacts/`, or app-specific files.
- **Telematics Control Module (TCU):** Logs telemetry and events: e.g., door locks/unlocks, ignition cycles, remote start commands, SOS call history, and possibly periodic location pings. Often has a SIM card slot and its own non-volatile memory for storing recent trip or status data. May record last known location and vehicle health data sent to manufacturer (e.g., monthly OnStar reports).
- **Navigation System:** Saved addresses ("Home," recent searches) and route history. Many nav units keep a tracklog or trip log (timestamped coordinates) for last trips. Files might be in proprietary formats or SQLite. Also, map update files on SD/USB can contain clues (e.g., last update time, or even leftover log files from the update process).
- **Event Data Recorder (EDR):** Typically embedded in the Airbag Control Module. Contains crash event snapshots: vehicle speed, brake status, accelerator, steering input, seatbelt, and more for ~5 seconds before impact. Only records when a crash-like event occurs. Data is locked in memory if airbags deployed, or overwritten after a few ignition cycles if not. Access via OBD or module connector; not in normal file system (requires EDR tool to retrieve).
- **Other vehicle ECUs:** ABS module may store last ABS activation timestamp, engine ECU stores DTCs (Diagnostic Trouble Codes) and freeze-frame sensor data at fault time, transmission ECU might log overheat events. These are typically accessed via a scan tool and not user-accessible, but pulling an image of the ECU's flash could reveal logs if analysis tools exist.
- **Smartphone (paired):** The driver's phone is a "device class" of its own – for Android Auto, Google Maps timeline (cloud) and local history, for Apple CarPlay, check iPhone backups for recent map destinations or Siri requests related to driving. Also, any companion apps (Tesla app, IoT device apps) on the phone store credentials and some usage logs. Include these in artifact collection.
- **IoT Home Assistant:** (e.g., Amazon Echo, Google Home) – activity (voice transcripts/history) is primarily cloud-side and visible via the user account; devices typically keep local configuration/state and limited logs (e.g., Wi-Fi settings, last activation time). The companion app exposes cloud history.
- **Wearables & Other IoT:** Smartwatches and fitness trackers record location tracks, heart rate, step count, etc., typically synced to a phone/cloud. Some data (last few days) can reside on the device itself in internal flash or memory. Smart security cameras/doorbells store recent video clips either on internal memory or removable media (SD card) along with logs of motion events and user accesses.
- In summary, identify each relevant device class in your case and pinpoint where its key artifacts live (on device vs companion device vs cloud). Vehicles concentrate data in infotainment/telematics; IoT spreads data across device, gateway, and cloud – you may need all to get the full picture.

5. Telemetry & Telematics

- Vehicles continuously produce telemetry (speed, steering, brake status, etc.) – telematics systems selectively record some of this data over longer periods. Unlike EDR's brief snapshot, telematics data can span months of normal driving at lower frequency. For instance, a telematics unit might log vehicle speed and GPS location every minute for fleet management or store last 100 trip summaries.
- OEM telematics services (GM OnStar, Ford Sync, Toyota Entune, etc.) often log events: remote door unlocks, horn honks, charge status (for EVs), tire pressures, and crash notifications. Much of this is sent to the cloud in real-time. Investigators can request backend logs (with legal process) to retrieve rich data like turn-by-turn routes, speed profiles, or even voice call recordings to call centers.
- Vehicles with advanced driver assistance (ADAS) or autonomy record specialized telemetry. *Example:* Tesla vehicles log detailed Autopilot data (accelerator, brake, steering angle, following distance alerts) and store it locally and in the cloud; NFI (Netherlands Forensic Institute) found Tesla logs can persist for >1 year on the car. These logs go beyond standard telemetry, capturing driver inputs and system states.
- Aftermarket or user-installed telemetry devices: Usage-Based Insurance (UBI) dongles track acceleration, braking force, time of day, etc. They usually upload data to insurer's servers but may keep a rolling buffer locally. Similarly, fleet telematics systems (in trucks, rentals) have internal storage for when the signal is weak – don't overlook pulling data from these devices directly.
- Accessing telematics data on the vehicle may require proprietary tools (e.g., some telematics ECUs can be imaged via JTAG or have debug ports). If direct extraction is not feasible, default to getting data from the service provider. Coordinate with the vendor if possible – some offer forensic data reports on request.
- Correlate telemetry with other sources: e.g., telematics speed vs EDR speed at time of crash (should match, differences may indicate data precision issues or tampering). Also use telemetry to fill gaps in other evidence: continuous GPS logs from telematics can link point-in-time evidence like a photo or transaction to a path of travel.
- Be aware of privacy features: some vehicles let users opt out of data collection or regularly purge data. If telemetry is crucial, verify if the owner had enabled or disabled such services. A lack of expected telematics data might itself be a clue (if deliberately disabled prior to an incident).

6. Bluetooth / Wi-Fi / Pairing Records

- Infotainment systems keep records of paired Bluetooth devices: typically the device name (e.g. “John’s iPhone”) and MAC address. This artifact links a specific phone to the vehicle. The pairing list may be accessible via the UI (settings menu) or stored in a config file or NV memory. Note the last connected date/time if available.
- When phones pair, many cars import the address book and call logs for in-car use. These imported contacts and call histories often remain on the head unit even after the phone is disconnected. Forensics can recover these (e.g., as a SQLite DB of contacts, or a call log table showing recent calls with timestamps and phone numbers).
- Vehicles equipped with Wi-Fi (hotspot or connectivity) will have Wi-Fi network info stored. Check for SSIDs the vehicle connected to (for OTA updates or user hotspot tethering) and devices that joined the car’s hotspot. The telematics module or infotainment might log when it connects to known Wi-Fi (e.g., home garage Wi-Fi) – useful for placing the car at a location.
- Artifacts to look for: *Bluetooth:* pairing database (often in a file like `paired_devices.bin` or within system settings), logs of connection events (some cars log “Bluetooth device X connected at 10:32pm”). *Wi-Fi:* configuration files (`wpa_supplicant.conf` on Linux-based systems containing saved Wi-Fi credentials), DHCP lease info for hotspot clients, or network event logs indicating when Wi-Fi was on/off.
- For IoT, pairing records are similarly important: smart hubs maintain lists of connected sensors (with unique IDs), and devices like smart TVs or speakers store Wi-Fi and Bluetooth pairings (e.g., a Google Home stores what phone was last used to set it up). These can place a person at a scene or show device interactions.
- Use pairing info in timeline correlation: e.g., if “Alice’s Phone” is paired to the car at 8PM and disconnected at 8:30PM, and an incident happened at 8:45PM, perhaps the phone left the vehicle – an investigative lead. Or if a victim’s phone auto-connects to a smart speaker at 10PM, it suggests they were home at that time.
- Also consider that clearing pairings is a common “anti-forensic” step when selling a car. However, traces might remain (in unallocated storage or logs). If you suspect a pairing was deleted, forensic carving of the infotainment image might still reveal the device name or MAC address fragments.

7. Navigation & Location

- Recover stored locations: most in-car GPS nav systems have a list of favorites (home, work, etc.) and recent destinations. These are prime evidence (e.g., suspect's saved addresses). They're usually stored in a nav database or flash memory. Examine any navigation app data for coordinates and place names.
- Pull route history if available: some systems log breadcrumb trails or trackpoints for trips. This could be a series of GPS coordinates with timestamps. For example, off-road vehicles or certain luxury cars have built-in trip recorders. Even if not user-facing, the data might be in a system log file. If found, map it out to visualize the path.
- Telematics location data: many telematics services keep a history of where the vehicle has been – either on the device or in the cloud. Last known location is often stored on the vehicle (for emergency services). Some devices log periodic pings (e.g., every ignition off event records location, or every hour). Querying the telematics module via OEM tools might retrieve these stored locations, or get them from backend records via warrant.
- Don't forget location metadata in other artifacts: images or videos from dashcams can have GPS coordinates, and paired phones often upload GPS data to cloud services (Google Timeline, Apple Location services). These can augment vehicle-sourced location info or fill gaps.
- Use mapping software to interpret data: if you get raw coordinates or address data, plot them in Google Earth or GIS tools to see the actual places. This can reveal patterns (e.g., regular visits to a certain address). Some tools can take a .GPX or .KML file from the car and directly show the route.
- When building timelines, align navigation events with time: e.g., "Destination input at 14:32 to 123 Main St" or "Vehicle arrived at last recorded point at 14:50". These can corroborate or refute alibis. Check if the car's clock was accurate – if not, adjust accordingly when comparing to other time sources.
- In cases of suspected tampering, see if navigation or GPS data was wiped. A sudden absence of any recent destinations (when other data is still present) could indicate a factory reset post-incident. However, map or POI data might still hold clues, like a cached tile or search term that survived.

8. Media & Communications

- **Call logs & contacts:** When a phone is paired, the car often downloads the recent call list and phonebook for hands-free use. These remain stored and are key evidence. Extract any call log database from the head unit – it shows who was called or who called, and when (with timestamps). This can mirror the phone's log but provides an independent record (especially useful if the phone isn't available or its log was wiped).
- **Text messages & emails:** Some infotainment systems display SMS or emails via connected phones (through MAP or other protocols). They might cache a portion of those messages. Look for text strings of known messages in a dump, or specific filenames (e.g., "msg_1234.txt") that could indicate stored communications. Even if content isn't stored, the system could log that a message was received at a certain time (with sender info).
- **Media files:** Check the storage for any audio, video, or images. Cars with built-in storage (hard drives or flash for music) may contain MP3s or other media files the user copied. Those files can be evidence (lyrics or content of interest, contraband media, etc.). Additionally, caches for album art or video thumbnails might exist (in folders like `/album_art/` or `/thumbs/`). These can reveal what media was played.
- **Voicemail and recordings:** If the vehicle has an in-car voicemail or memo recording feature (some premium systems do), retrieve those audio files. Also, some telematics (e.g., OnStar) may record calls with operators; if accessible, those recordings can be critical evidence (though often held by the provider rather than stored in the car).
- **Voice assistant logs:** Modern cars and IoT devices often have voice control. Check for logs of voice commands (even if audio isn't stored). For example, a smart car might log "Voice command: 'Call Mom' at 18:00". Alexa/Google Home devices keep cloud logs of questions asked – obtain these via the user's account or subpoena, as they can reveal intent or timeline ("User asked smart speaker for the time at 3:15 AM", etc.).
- **In-car internet and comms:** Some infotainment systems have web browsers or apps (Facebook, Twitter, WhatsApp integration). If enabled, they could store cookies, chat logs, or browsing history in the system. Check for any apps on the system (the SWGDE categories list things like Weather, Facebook data). For IoT, consider chat logs or notifications (e.g., a smart fridge might log that a message was displayed on its screen).
- Cross-device communication: For vehicles with connected apps (like sending a destination from phone to car, or remote text messaging via car), evidence might be split – part on phone, part on car. Make sure to gather both sides. E.g., a car might show an SMS was sent using the vehicle interface, while the phone has the message content. Combined, they give a full picture.

9. EDR / Crash Data

- EDRs (“Black Boxes”) capture critical crash metrics in a short window. Historically ~5 seconds pre-crash and a few seconds post-impact; note newer NHTSA rules extend required pre-crash recording to 20 s at 10 Hz on applicable new vehicles. Key data elements include: vehicle speed, engine RPM, throttle position, brake application (on/off), seatbelt status, airbag deployment time, and crash severity (delta-V). Data resides in the airbag control module’s memory.
- Trigger conditions: A recording happens when a crash event meets certain thresholds. There are usually two types of events logged: airbag deployment events (which are stored and locked) and non-deployment events (hard braking/near-crash that didn’t fire airbags). Non-deployment event data can be overwritten by subsequent events or after a set number of ignition cycles. If investigating a minor crash, gather EDR data quickly before it’s overwritten.
- Data retrieval: Use a Crash Data Retrieval (CDR) tool (Bosch CDR is the standard) to download the EDR report via OBD-II or directly from the module. The tool will output a report (PDF or similar) of the recorded data. Ensure the vehicle you have is supported by the tool (consult the supported vehicle list; e.g., many post-2010 cars in US, but coverage varies by make/year).
- No timestamps in EDR: EDR data is relative timing (t = 0 at impact). It does not record absolute date/time. So you must correlate the EDR event with other evidence (like 911 call time, or vehicle infotainment clock) to confirm which incident the data corresponds to. Be cautious not to assume the most recent event is the incident without correlation.
- Legal considerations: EDR data is often legally protected. In the US, the vehicle owner owns the data by law; you need owner consent or a warrant to retrieve it. Some jurisdictions also have specific statutes about EDR. Document the consent/warrant clearly in your case notes and evidence logs.
- Handling EDR modules: If you cannot download on-scene (e.g., vehicle power is compromised), you may remove the airbag control module and image it later. However, removal can be dangerous (disconnect battery and wait to avoid airbag deployment risk). If removed, treat it gently (crash data is non-volatile but modules can be damaged). Label the module with orientation (important for delta-V interpretation) and avoid exposing it to static or moisture.
- Interpreting the report: Understand each parameter – e.g., “Pre-crash speed 1.5s before impact: 45 mph, Brake switch: ON”. Pay attention to any anomalies or error codes reported in the EDR data (some reports flag if data couldn’t be recorded fully). Use an expert if needed to analyze crash data in context of accident reconstruction.

10. CAN Bus & Vehicle Networks

- Modern vehicles have multiple networks: usually at least two CAN buses (high-speed CAN for powertrain, medium-speed for body/infotainment) and often LIN subnets (for low-speed components like seat sensors), plus others (FlexRay or automotive Ethernet in newer cars for ADAS). The OBD-II port typically gives access to the powertrain CAN (and sometimes a gateway to other buses).
- **Capturing CAN traffic:** If investigating a vehicle cyber incident or need to reproduce events, you can sniff CAN data using interfaces (Peak CAN, CANtact, etc.) and tools like Wireshark or SavvyCAN. However, live capture requires powering the car on (which may change state and generate new messages). Avoid live sniffing on the evidence vehicle unless necessary. Instead, you might use a similar vehicle or bench setup to understand message patterns.
- **Diagnostic data via CAN:** Use an OBD-II scanner or forensic scan tool in read-only mode to pull DTCs (Diagnostic Trouble Codes) and freeze-frame data. DTCs are stored snapshots when an error occurred (e.g., engine over-temp at 1200 RPM on date X). These can provide timeline info and indicate anomalies (e.g., crash event might trigger DTCs in airbag or fuel cutoff modules).
- Some vehicles log certain CAN events: e.g., a body control module might log the last time a door was opened or when anti-theft alarm was triggered. These logs are in EEPROM on the module and require specialized tools (often dealer tools) to read. If available, they can be gold for timeline (e.g., “Driver door opened at 10:32:10 PM”). Investigators should liaise with OEM or use advanced diagnostics to get these.
- When analyzing CAN logs (if you have them), use known databases (DBC files) for the vehicle if available to translate message IDs to human-readable signals. Without OEM info, some common signals can be inferred (e.g., speed often appears on CAN as a two-byte value scaling by 0.01). Community forums or prior research may have reverse-engineered portions of the CAN for popular models.
- **Other networks:** If pertinent, be aware of MOST (Media Oriented Systems Transport) for audio/video, which might carry evidence (like a video feed from cameras) but is not typically logged; and automotive Ethernet (used in modern vehicles for high data throughput, e.g., driving data recorders in autonomous cars). These usually don’t have easy forensic taps; you rely on whatever the car recorded from them, if anything.
- In summary, CAN bus forensics is usually indirect: rather than raw captures, you get processed data (logs, DTCs, event flags) stored by ECUs. Focus on retrieving those high-level artifacts. Direct CAN analysis is more common in cybersecurity investigations (looking for anomalies or attack traces) than in typical crime scene forensics, but it’s good to know how if needed.

11. Acquisition Methods

- Determine the least invasive method to get the data you need. Whenever possible, perform a logical acquisition using vendor interfaces (to avoid ripping out hardware). For many infotainment systems, this means using specialized cables or connectors to extract data via diagnostic or debug ports. E.g., some head units can be imaged by simply plugging a forensic tool into the OBD-II or USB port (if supported by tools like iVe).
- If logical access isn't available, plan for a physical acquisition. This could involve removing the unit and doing a chip-off or direct eMMC flash extraction. Tools like heat guns, chip readers, or in-circuit programming devices (JTAG/SPI) come into play here. Be aware of soldered-down storage and possible encryption – physical imaging might yield an encrypted blob if keys are hardware-bound.
- Use manufacturer service modes when available: some infotainment systems have maintenance menus or key combos that allow data export or clone (primarily for updates). Example: certain Toyota/Lexus systems allow saving data to USB in dealer mode. If you have access, leverage these instead of destructive methods.
- Leverage commercial forensic tools: Berla iVe is a leading vehicle forensics kit that provides both software and hardware to acquire data from dozens of car makes/models. It includes custom interface boards for direct-to-PCB connections (when needed) and supports physical and logical acquisitions with built-in parsing. Other tools like Cellebrite or MSAB may offer limited vehicle support (often by extracting from connected mobile apps or known filesystems, not as comprehensive as iVe).
- For EDR, the Bosch CDR tool is the go-to. Ensure you have the correct cable for the vehicle (DLC or direct-to-module adapters). Always perform EDR downloads in a read-only manner (the tools generally do). Print the EDR report or save it securely; that is your "forensic image" of crash data. If multiple events are stored, download all if possible.
- IoT device acquisition can vary: try standard approaches first (e.g., Android-based IoT – use ADB backup or rooting to get an image; for others, see if the vendor provides a web interface or API to download data). If not, resort to hardware: open the device, identify storage chips (NAND, NOR flash), and either use test pads for JTAG/SWD or remove the chip for reading in a programmer. Document any such process meticulously (photos of disassembly, etc.).
- Cloud data acquisition: Many IoT/vehicle services provide user data portals or GDPR downloads. For instance, Tesla owners can request their vehicle data dump; Amazon allows downloading Alexa voice history. Use these legal channels when possible – they can save time and provide well-formatted data. Just ensure you have legal authority or user consent to use account credentials for this purpose.
- Always hash and preserve original data images: Whether it's a 8 GB infotainment flash image or a 1 GB IoT firmware dump, compute a SHA256 hash and secure the original. Do analysis on copies using write-blockers or mount read-only. For live acquisitions (like pulling data via a tool), document the process and any tool output (some tools will generate a report – include that in evidence documentation).

12. Volatile Data & Live Response

- Recognize what data is volatile: in vehicles, RAM contents (running processes, unsaved logs, decryption keys), and in IoT, ephemeral sensor data or network connections. Once power is lost or the device reboots, this information is gone. If live response is feasible, prioritize capturing this data. For example, if an infotainment system is running and you can access a developer console, dump the memory or at least run commands to gather RAM-based info (process list, open network sockets, etc.).
- Use safe execution: booting from external media on a vehicle system (if supported) or connecting via serial console can allow memory dumps without altering storage. E.g., some QNX systems might allow a ` dumpram` utility or similar. In IoT Linux devices, you might use `dd if=/dev/mem` or `/proc/kcore` (with caution and appropriate tools) to capture memory, but note that on modern Linux kernels this may be restricted.
- If you suspect malware or an intrusion (vehicle cyber attack scenario), live memory is crucial for indicators (malicious processes, network sniffers, etc.). Utilize memory forensics tools like Volatility or Rekall on any RAM image obtained to search for known signatures or suspicious strings.
- Volatile data extends to network state: for a connected car, what cell towers is it connected to at the moment? What is its IP address? For IoT, what MQTT broker or cloud server is it actively connected to? A live capture of these can later help connect the dots (e.g., a car was connected via IP X which geolocates to region Y at time of seizure). Use `netstat` or similar on devices if possible (many IoT and embedded OS have busybox netstat or ifconfig to show connections).
- Plan for safe shutdown after volatile capture: once you've gathered what you can, shut down the device properly to preserve file system integrity (for cars, follow the earlier shutdown procedure; for IoT, use the OS shutdown command if available). This ensures any buffered data is written to disk and reduces corruption.
- For devices you cannot directly interact with (no interface or locked), your only chance for volatile data might be preserving power until a specialist can attach a debugger. In such cases, transport the powered device in a Faraday enclosure to the lab (with a portable power supply if needed). This is complex, but for some IoT (like a powered-on encryption device), it might be the only way to retain keys in RAM.
- Always weigh the risk of live response: doing so on an unfamiliar system could trigger self-wipes or crashes if not careful. If documentation or experience is lacking, it may be safer to preserve the device as-is for lab analysis rather than executing random commands. However, not capturing volatile data is a missed opportunity that cannot be undone – so make an informed decision quickly at triage time.

13. Timeline Analysis

- Building a timeline is essential in vehicle/IoT investigations. Include data from the vehicle, IoT devices, and external sources in one chronological sequence. For each event, note the source. For example: “2025-08-01 14:32:15 – Car ignition ON (infotainment log)”, “14:32:50 – Driver’s phone connected via Bluetooth (vehicle BT log)”, “14:35:00 – Smart home camera detects motion at garage (IoT camera log)”. This helps illustrate the story of what happened and when.
- Normalize timestamps to a common reference (e.g., UTC or local time) and clearly state the offset if needed. Vehicles might use GPS time (which is essentially UTC with no leap seconds) or local time set by the user, which could be wrong. IoT devices often use NTP (internet time) but could be off if disconnected. If a car has a GPS time sync event (many log when they last synced clock), use that to correct the timeline if necessary.
- Use graphical timeline tools or spreadsheets to visualize overlapping events. You might find, for example, that a car’s door-open event aligns with a specific IoT event at a house (like a smart door unlocking). Or that an EDR-recorded crash at 10:05:30 aligns with a 911 call at 10:06 and a smart watch detecting a fall at 10:07. Converging multiple data points solidifies conclusions.
- Identify gaps and silences: a timeline might show no data from 12am to 4am for the vehicle (perhaps it was off), but an IoT thermostat recorded movement at 2am – indicating someone was around. Gaps might also indicate device tampering (e.g., dashcam turned off). Mark these gaps and try to explain them with context.
- Back up timeline assertions with evidence: each entry in your timeline should trace to an artifact (log file, database entry, video timestamp). This is where meticulous documentation during analysis pays off. If you note “Engine turned on at 7:45PM”, be ready to show the log line or data source that confirms it. This is crucial for court presentations.
- Keep time zones in mind if the investigation spans regions (vehicle clock vs cloud data vs CCTV might all be in different zones). Convert and label times clearly. If daylight savings or timezone shifts occurred in the timeline period, account for those to avoid confusion.
- As a best practice, have another investigator peer-review the timeline. Fresh eyes might spot an event you missed or mis-ordered. Timelines can become complex, but they are one of the most powerful tools to communicate the sequence of events in a case involving multiple data sources.

14. Tools & Parsers (open-source & commercial)

- **Infotainment/Telematics Data Extraction:**
  - *Open-source:* No dedicated open tool for comprehensive car data extraction (most techniques are manual). Use general forensic tools: **Autopsy/The Sleuth Kit** (to carve and parse images), **binwalk** (to extract filesystem from firmware) for do-it-yourself extractions.
  - *Commercial:* **Berla iVe** – purpose-built kit for vehicle forensics (supports physical/logical acquisition for many makes/models, with built-in parsers). Others include **Cellebrite UFED** and **MSAB XRY** (limited vehicle support, sometimes via pulling phone data or specific modules), and **Magnet AXIOM** which can ingest vehicle-derived data for analysis (but relies on you acquiring the data first).
- **EDR (Crash Data) Retrieval:**
  - *Open-source:* None – EDR retrieval is proprietary by design. At best, researchers have Python scripts to parse certain raw EDR binary files, but you typically won't get those without the proper tool.
  - *Commercial:* **Bosch Crash Data Retrieval (CDR) Tool** – industry standard hardware/software to download EDRs. Also, **Crash Data Group** sells kits and software (often rebranding Bosch). Some OEMs (Tesla, etc.) have their own tools used internally – not generally available.
- **CAN Bus & Network Analysis:**
  - *Open-source:* **SocketCAN** (Linux kernel framework) with tools like **candump** for capturing CAN traffic; **Wireshark** with CAN plugins to analyze captured .log or .pcap files; **SavvyCAN** (GUI tool for CAN logging and DBC applying); **cantact** or **CANalyzat0r** hardware (open hardware interfaces) for connecting to OBD-II.
  - *Commercial:* **Intrepid VehicleSpy** (powerful CAN analysis and injection suite), **Vector CANoe/CANalyzer** (engineering tools for in-depth CAN/LIN/Ethernet simulation and logging), **Peak PCAN** tools, and **PLC Hunter** (for industrial vehicles). These are more for specialists and require CAN bus knowledge.
- **IoT & Embedded Forensics:**
  - *Open-source:* **Firmware analysis tools** like binwalk, **FirmWalker** (scans extracted firmware for creds/artifacts), **Volatility** (if you get memory dumps), and **Foremost/Scalpel** for carving data from flash dumps. Also, general tools: **FTK Imager** (free) to image SD cards/USB from devices, and **GNU ddrescue** for cloning flash memory.
  - *Commercial:* **Magnet AXIOM & IEF** (IoT artifact parsing modules, support for cloud data from IoT services), **Oxygen Forensic Detective** (has IoT data parsers and cloud extractors for some platforms), hardware like **ACES Laboratory JTAG adapters** or **Atola** for chip-off reading. Also, **BlackLight** by BlackBag can parse APFS and other filesystems if your IoT device uses those (e.g., some smart gadgets running iOS/tvOS). These tools help parse and organize data after you acquire it.
- **Analysis & Correlation:**
  - *Open-source:* **Plaso/Log2Timeline** (timeline creation from multiple log sources), **Elastic (ELK) Stack** for aggregating logs (could ingest car and IoT logs to query), **Timeline Explorer** (free GUI for timeline CSVs). For mapping, **Google Earth Pro** (free) to import GPS coordinates.

- *Commercial:* **Maltego** (can be used to link entities like devices, locations, people in a case graph), **IOD digital investigation software** (some have modules for vehicle data), and general suites like **Nuix** or **EnCase** which can index large sets of data from various sources for cross-reference. These can help when you have thousands of data points from car and IoT dumps.

15. Anti-Forensics & Pitfalls

- **Factory resets & data wipes:** A quick reset of the infotainment or IoT device can delete user-visible data (paired devices, call logs, nav history). Always check if a reset was done – for example, if the car has only factory default profiles and no phone paired, yet the owner clearly used it, suspect data wiping. Use forensic carving on the image to try to retrieve deleted SQLite records or files.
- **Volatile data loss:** If the vehicle battery was pulled or IoT device powered off improperly, some data may have been lost before capture. Volatile caches (e.g., a telematics buffer of recent coordinates in RAM) would vanish. That's why following proper shutdown steps is key. Consider what might have been in RAM and see if remnants exist elsewhere (for instance, some systems periodically commit RAM data to disk – the interval matters).
- **Log overwrites & circular buffers:** Many automotive logs are circular buffers with fixed size (e.g., only last 50 trips, last 100 Bluetooth connections). If the suspect drove extensively or paired many devices after the event, earlier evidence might have been overwritten by the system design. Telemetry dongles might overwrite old trips once memory is full. Always grab data as soon as possible, and be mindful that "absence of data" could mean it was overwritten naturally.
- **Time sync issues:** If the device clocks are off, you could mis-order events. A car not synced to GPS might have an incorrect clock by minutes or hours. IoT gadgets might be on UTC while your other data is local time. Double-check any timeline by verifying clock offsets (e.g., look at a known event like a phone connection – the car log vs phone log time difference reveals any clock drift).
- **Encryption and access control:** Some infotainment systems encrypt user data or require a PIN (e.g., Tesla's internal storage is encrypted, requiring decryption keys that are not trivial to get). IoT devices might enforce secure boot or encrypted storage (some doorbell cams encrypt video on SD cards). If you encounter encryption, you may need to capture keys from RAM (hence live response importance) or use manufacturer help. Trying brute force or chip swapping could trigger tamper mechanisms.
- **Remote interference:** Once it's known a vehicle or device is evidence, a savvy suspect might attempt remote access to tamper (e.g., using the vehicle's connected app to wipe data or an IoT device's cloud link to delete logs). That's why isolation is emphasized – a lesson learned from cases where data "mysteriously" disappeared. Ensure the vehicle/IoT can't phone home after seizure.
- **Proprietary data misinterpretation:** Without clear documentation, it's possible to misread data (e.g., interpreting a hex value as speed when it's actually something else). Cross-verify important findings with a second source. For example, don't conclude "airbag was off" from one byte in a dump unless you're certain of the format. Consult manufacturer data or community research for that specific model to avoid mistakes. Data storage in vehicles is the "wild west" – each manufacturer, and even model year, can store things differently.
- **Hardware booby traps:** Rare but possible – some devices might have anti-tamper (e.g., zeroizing keys if opened). High-end aftermarket trackers or certain secure IoT devices could do this. If you suspect this, go slow (shield from RF, keep power stable) and consider consulting hardware experts. Also, be careful with airbag modules (not for data loss, but safety – they can deploy if mishandled).
- Lastly, maintain healthy skepticism. If something seems too empty or too "perfect" (like a complete lack of any

personal data in a used car), consider the likelihood of deliberate wiping. Conversely, don’t assume malicious intent for every anomaly – sometimes logs drop entries or devices glitch. Use corroboration to distinguish between technical quirks and tampering.

16. Legal & Chain of Custody Notes

- **Authority to seize and examine:** Always confirm you have the legal right to access the data. Vehicles and IoT devices contain personal data, so warrants or owner consent are usually required. Note that in many places, the car's EDR data legally belongs to the vehicle owner – a warrant or written consent is a must before retrieving it. Similarly, data from an IoT cloud account requires proper legal process (e.g., warrant or provider disclosure process).
- **Documentation:** From the moment of seizure, document every action. For a vehicle, record the make, model, VIN, and condition (doors locked/unlocked, ignition state). If the whole vehicle is seized, log where and how it's stored (secure impound, etc.). If individual modules are removed, label them with identifiers and their location in the vehicle. Use evidence tags on hardware and anti-static bags for circuit boards.
- **Chain of custody forms:** List each digital device or component separately (e.g., "Item 7: Infotainment control module, serial number XYZ, removed from vehicle ABC at scene"). Track transfers – when it goes to lab, to a forensic examiner, to court, etc., with signatures. Because IoT devices can be small, double-check that all pieces (cables, power adapters which might store info like pairing codes) are accounted for and listed.
- **Privacy & scope:** Be mindful to stay within the scope of your warrant. If your warrant is for vehicle data related to a specific incident, avoid poking into unrelated personal data (e.g., unrelated past trip locations) unless it's explicitly covered or immediately relevant. Over-collection can lead to challenges in court. If you find evidence of other crimes (say, the infotainment has child exploitation images as album art cache), involve legal counsel to determine next steps before proceeding further on that tangent.
- **Collaborate with specialists:** Some data (like EDR or telematics proprietary info) may require expert interpretation. Consider getting an expert witness or contacting the manufacturer (some have forensic liaison programs) to validate your extraction method and findings. This can pre-empt defense arguments about reliability.
- **Handling physical/digital evidence interplay:** If the investigation also requires latent prints or DNA from the car, coordinate so those exams happen without spoiling the digital evidence. The Interpol guide suggests preserving digital state before extensive physical processing. Likewise, ensure physical exam (e.g., pulling a fingerprint off the screen) doesn't require powering the system in a way that alters data.
- **Reporting:** In your forensic report, clearly explain how the data was obtained from these non-traditional sources. Jurors or judges may not be familiar with vehicle or IoT forensics. Document the steps (e.g., "extracted vehicle data using Berla iVe, which utilizes manufacturer-specific protocols to ensure read-only acquisition") and attach tool logs or output reports. Transparency will help defend against Daubert/FRYE challenges regarding your methods.
- **Retention and Return:** Vehicles and large IoT items might be returned to owners before trial due to storage costs. Make sure comprehensive images/copies are made of all digital evidence so analysis can continue after the item is released. For any cloud data, once obtained, preserve it offline (don't rely on continued access to the account). Ensure legal return of property is documented (and any forensic hardware like a tapped wiring harness is removed from the vehicle).

17. References & Mapping

- **SWGDE Best Practices (Vehicles & IoT):** Refer to SWGDE documents for guidance. e.g., *SWGDE Best Practices for Vehicle Infotainment and Telematics Systems* (v2.0, 2016) – it outlines data types available and proper handling. Also *SWGDE Best Practices for IoT Seizure and Analysis* (v1.0, 2023) for procedures on identifying and collecting IoT devices. These were used to inform many steps in this cheat sheet.
- **NIST & Interpol Guidelines:** See NIST publications and the INTERPOL *Guidelines for Digital Forensics First Responders*, which include sections on Automotive (section 5.12) and general handling that we've summarized (e.g., proper shutdown sequence, evidence handling and isolation).
- **Data mapping resources:** Use manufacturer-specific forensic guides if available. For example, Tesla's decrypted log info was published by the Dutch Forensic Institute – knowing what Tesla logs helps target analysis. Community sites (XDA, etc.) have file system maps for systems like Uconnect (e.g., showing `/fs/mmc0/` usage). Always verify with a test unit when possible.
- **Cross-reference artifact locations:** Map each artifact to its potential sources to ensure comprehensive collection. For instance:
  - **Phone call** – could be in infotainment call log DB, the phone itself, and possibly telematics if an emergency call.
  - **Location at time X** – could come from nav system tracklogs, telematics backend, phone GPS, or even IoT devices (like a paired smartwatch GPS).
  - **Door open event** – might be logged in car's body control module and also inferred from smart home data (garage door, house entry sensor).

  Using a matrix approach (Artifacts vs. Sources) can be helpful during your analysis planning.
- **Training and continuous learning:** Vehicle and IoT forensics is evolving. Stay updated via courses (SANS VEH **[Vehicle Forensics]**, blackhat presentations), forums, and publications (DFRWS Automotive workshop papers, IEEE publications on automotive forensics). Also, follow tool updates (Berla iVe release notes often list new vehicle support, which hints at new data types or locations being tackled).
- **Final mapping advice:** Always think in terms of the ecosystem. Vehicle forensics isn't just the car – it's the car, the key, the phone, the cloud, the home. IoT forensics isn't just the gadget – it's the gadget, the network, the hub, the user's phone. Map out that ecosystem early in the investigation to guide what data to collect from where. This cheat sheet provided examples and common locations, but each case will have its unique map of data – draw it out, then execute your collection and analysis plan based on that map.